Hybrid genetic algorithm using a parametric method to solve the two-dimensional phase unwrapping problem

Salah Karout, Munther A. Gdeisat, David R. Burton, Michael J. Lalor
CEORG, General Engineering Research Institute,
Liverpool John Moores University, UK,
s.a.karout@2004.ljmu.ac.uk

Abstract

A hybrid genetic algorithm is proposed to solve the two-dimensional phase unwrapping problem by minimizing the \( L^p \)-norm between the unwrapped and wrapped phase gradients. The unwrapped phase is approximated by estimating the parameters of a function chosen by the genetic algorithm that can achieve the global minimum of the \( L^p \)-norm. Different predefined functions are used on the basis of the object characteristics being unwrapped whether continuous and discontinuous object. The \( n^{\text{th}} \)-grade polynomial is one of the functions used to approximate the unwrapped phase. The hybrid genetic algorithm uses a polynomial weighted least-squares phase unwrapping solution as an initial approximation of the unwrapped phase map to speed up convergence to global optima. The algorithm is robust in unwrapping very noisy wrapped phase maps. It is computationally efficient and very fast. The performance of the algorithm is tested using simulated and real wrapped phase maps and results are compared to that of the existing two-dimensional \( L^p \)-norm phase unwrapping algorithm.

Key words

Phase unwrapping, Genetic Algorithm, weighted least-squares and polynomial regression.

1 Introduction

Many digital image processing techniques may be used to extract phase distributions from images (e.g., fringe pattern or magnetic resonance scan images or Synthetic Radar Interferometry images) in order to obtain the information embedded within the phase map (e.g., height, magnetic field, velocity and displacement). Such techniques include Fourier fringe analysis, phase stepping and the wavelet transform method [Huntley et al., 2001]. These methods of calculating the phase distribution suffer from one disadvantage, which is the use of the arctangent operator to extract the phase distribution. The arctangent operator produces results wrapped onto the range \(-\pi\) to \(+\pi\). Thus, in order to retrieve the continuous form of the phase map, an unwrapping step has to be added onto the phase retrieval techniques [Cusack et al., 1995]. This unwrapping step is not a straightforward technique because of the presence of noise, object discontinuity and the violation of Shannon’s law due to undersampling in real wrapped phase maps. As a result, many phase unwrapping algorithms have been developed to solve this problem. However, the variety of forms, shapes and densities of noise that might be found in real wrapped phase maps makes the problem of phase unwrapping complex and difficult to solve, even given the significant amount of research effort expended to date and a large number of exiting phase unwrapping algorithms.

1.1 Phase Unwrapping

Phase Unwrapping (PU) is a technique used on wrapped phase images to remove the \( 2\pi \) discontinuities embedded within the phase map. It detects a \( 2\pi \) phase jump and adds or subtracts an integer offset of \( 2\pi \) to successive pixels following that phase jump based on a threshold mechanism. The threshold mechanism states that if the phase difference between two successive pixels in a path \( P \) as in Eq. (1):

\[
\Delta \Phi(p_i) = \Phi(p_i) - \Phi(p_{i-1})
\]

is greater than \(+\pi\), subtract a \( 2\pi \) offset to all successive pixels in the path. However, if the phase difference is less than \(-\pi\), add a \( 2\pi \) offset to all successive pixels in the path. Then, by locating all discontinuities in the wrapped phase map, the phase at every pixel will change by an integer \( k \) multiples of \( 2\pi \) depending on the pixel position in the unwrapping path. This can be summarised by the wrapping operator in Eq. (2):

\[
W[\Phi(p_i)] = \Phi(p_i) + 2\pi k(p_i)
\]

\[k(p_i) \in \mathbb{Z}\]

where \(-\pi \leq W[\Phi(p_i)] \leq +\pi\). The wrapping operator could be modified to specify the gradient of
the corrected phase difference between two successive pixels in the unwrapping path and it is stated in Eq. (3):

\[ \nabla \Phi(p_i) = W[\Phi(p_i) - \Phi(p_{i-1})] \quad (3) \]

A simple global method of phase unwrapping is to minimize the distance between the phase gradient estimate (unwrapped phase) and the true gradient as presented in Eq. (4) [Ghiglia et al., 1998]:

\[
\varepsilon^p = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| \psi_{i+1,j} - \psi_{i,j} - \nabla \phi^x_{i,j} \right|^p + \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \left| \psi_{i,j+1} - \psi_{i,j} - \nabla \phi^y_{i,j} \right|^p 
\]

where \( i \) and \( j \) are indices of the pixel location in the image respectively,

- \( \phi_{i,j} \) is the given wrapped phase,
- \( \psi_{i,j} \) is the approximated unwrapped phase,
- \( \nabla \phi^x_{i,j} = W[\phi_{i+1,j} - \phi_{i,j}] \)
- and \( \nabla \phi^y_{i,j} = W[\phi_{i,j+1} - \phi_{i,j}] \)

are the wrapped phase gradients in \( x \) and \( y \) directions respectively.

\( M \) and \( N \) are the size of the phase map image.

Two major classes of phase unwrapping algorithms are path-following and least-square methods. The path-following methods deal with the problem of residues directly by identifying the residues and eliminating their presence in the phase map by balancing their polarities and branch-cutting them from the phase map. Once the residue's effect eliminated, phase unwrapping can take any path through the phase map and branch-cut act as barriers to unwrapping corrupted areas in the phase map. There have been many path-following algorithms developed which vary in the way this method is implemented.

On the other hand, least-squares methods are completely different than path-following methods. They are divided into three different types: unweighted least-squares, weighted least-squares and \( L^p \)-norm methods. These methods in general minimize up to a certain degree (least-square to the degree order and \( L^p \)-norm raised to the \( p \) degree order) the difference between the gradients of the wrapped and the gradient of the unwrapped solution in both \( x \) and \( y \) direction. However, these methods do still indirectly deal with the residue problem because their solution is obtained by integrating over the residues to minimize the gradient differences [Ghiglia et al., 1998]. Least-square methods have the advantage that it they are more noise tolerant and they achieve the global smoothness of the unwrapped solution.

The unweighted and the weighted least-squares methods are well defined mathematically and equivalent to the solution of Poisson’s partial differential equation which leads to a sparse linear equation that can be solved by any iterative method. However, these methods generate a very large number of linear equations to be solved equivalent to the total number of pixels in the phase map. They have a very large computational time and their convergence rate is very critical to a successful solution. Moreover, weighted least-squares requires weights to achieve better results than the unweighted counterpart. These weights are user defined weights generated from quality-maps used to isolate corrupted areas with residues by masking them out of the wrapped phase data to diminish their effect on the unwrapped solution. A drawback to these two methods is if some residues where not masked out, they will cause the unwrapped phase to be severely corrupted depending on the density of the unmasked residues. A more advance method developed by Ghiglia et al. [Ghiglia et al., 1998] is \( L^p \)-norm which uses similar methods like the two previous least square methods to solve the phase unwrapping problem. However, this method does not compute the minimum \( L^2 \)-norm but the general minimum \( L^p \)-norm. In essence, by computing the minimum \( L^p \)-norm where \( p\neq2 \); this method can generate data dependent weight unlike the weighted least-square method. The data-dependent weights can eliminate iteratively the presence of the residues in the unwrapped solution. Unfortunately, this method is more robust than the previous mentioned least-squares method but it is more computationally intensive.

In this paper, a global phase unwrapping algorithm is proposed that uses a genetic algorithm to estimate the parameter coefficients of an \( n \)-order polynomial used to create the unwrapped phase solution that minimizes the \( L^2 \)-norm error between the gradient of the solution and the gradient of the wrapped phase map. This method is similar in concept to least-square and \( L^p \)-norm phase unwrapping methods developed by Ghiglia et al. [Ghiglia et al., 1998] except it does not rely on the wrapped phase data to construct the unwrapped solution. However, it uses a polynomial to construct the unwrapped surface solution. In essence, it has a major advantage on least-squares and \( L^p \)-norm methods by being free from any residue disturbances generated from the wrapped phase map. In other words, the residues embedded in the wrapped phase cannot affect the unwrapped phase solution. The reason is because the wrapped and the unwrapped phase maps are completely independent from each other. The only
A faster way to learn the optimal search path from experience [Willi-Hans et al., 2002]. Genetic algorithms are specifically designed to solve non-deterministic polynomial problems called NP-hard problems which involves large search spaces containing multiple local minima [Willi-Hans et al., 2002]. It has also been applied to many combinational optimisation problems and has proved its robustness and speed.

Genetic algorithm iterations consist of a set of chromosomes where every chromosome represents a candidate solution for the problem to be solved. Chromosomes consist of a number of genes that can be considered as variables to the problem. Manipulating these genes (variables) will result in creating new solutions. To evaluate how much the solution has been improved, a fitness function (evaluation function) is tailored to the problem.

Such a method uses three natural techniques:

- **Natural selection**, which allows the best genes (e.g., healthy in human terms) to be selected thus giving them more possibility of moving on to the next generation,
- **Crossover**, which is responsible for producing new chromosomes (offsprings) from the original chromosomes (parents),
- **Mutation**, applies deliberate changes to a gene at random, to keep variation in genes and increasing the probability of not falling into a local minimum solution. It involves exploring the search space for new better solutions.

| Randomly generate initial population, size Pop, Evaluate the fitness of every chromosome, |
| Do                                                                                   |
|  For chromosome = 1 to Pop/2                                                         |
|  Use selection operator to select to chromosomes,                                    |
|    If rand_no. < P_s                                                                  |
|    Apply the crossover operator,                                                     |
|    If rand_no. < P_m                                                                  |
|    Apply the mutation operator,                                                      |
| End                                                                                    |
| Replace the new population with the old population,                                  |
| Evaluate the fitness of all new chromosomes,                                        |
| Preserve the most-fit chromosomes so far and copy into the new population,           |
| Repeat while not converged or not at maximum iteration.                               |

Figure 1: A pseudo-code of a general genetic algorithm; where $P_s$ and $P_m$ are the probability of performing crossover and mutation respectively.

## 3 HGA for Parameter Estimation

The proposed algorithm as mentioned previously relies on estimating the parameters of an $n^{th}$ order-polynomial to approximate the unwrapped surface solution from the wrapped phase data. The proposed algorithm uses a genetic algorithm to obtain the coefficients of the polynomial that best unwrap the wrapped phase map. However, by providing the genetic algorithm with an initial population created by randomly choosing a number between two limits will cause the algorithm to take a very long time to converge to the global optimum solution. A faster way was achieved by obtaining an initial solution that is used to create the initial population.

In this proposed algorithm, the complexity of the problem relies on the order of the polynomial used to reconstruct the unwrapped surface solution. By increasing the order of the polynomial, more precision...
and a lower minimum $L^p$-norm error are achieved. The number of coefficients of the polynomial also increases with the order of the polynomial.

The proposed algorithm is summarised in Fig. 2:

- Calculate the quality map of the wrapped phase map using maximum phase gradient quality map [Ghiglia et al., 1998].
- Unwrap the wrapped phase using quality guided algorithm [Ghiglia et al., 1998].
- Surface-fit the unwrapped solution with a polynomial using weighted least-square multiple regression controlled by the quality map weights.
- Coefficients of the surface-fitted polynomial are given to the genetic algorithm as an initial solution to lower execution time.
- Genetic Algorithm minimizes the $L^p$-norm error between the gradient of the polynomial unwrapped surface solution and the gradient of the original wrapped phase map.

Figure 2: A summary of the proposed parameter estimation genetic algorithm.

### 3.1 Coding the Phase Unwrapping problem in GA Syntax Form

Any optimisation problem using a GA requires the problem to be coded into GA syntax form, which is the chromosome form. The chromosome can be used to represent a graph or an equation or a system. In this problem, the chromosome consists of a number of genes where every gene correspond to a coefficient in the $n^{th}$-order surface fitting polynomial as described in Eq. (5) and Fig. (3).

$$f(i, j) = a_0 + a_1 i + a_2 j + a_4 i^2 + a_4 ij + a_5 j^2 + \ldots + a_{\frac{(n+1)(n+2)}{2}} i^n$$

(5)

| $a_0$ | $a_1$ | $a_2$ | $a_3$ | $\ldots$ | $a_{\frac{(n+1)(n+2)}{2}}$ |
|-------|-------|-------|-------|-----------|

Figure 3: Coding scheme of the coefficients of the $n^{th}$-order surface fitting polynomial into the chromosome syntax form.

### 3.2 Initial Population

A GA requires an initial population of chromosomes where each chromosome represents a possible solution. From this initial population, the GA starts using a stochastic search to achieve the global optimum solution. The method that is used to create the initial population will determine the speed of convergence to an optimum solution, as well as the size of the population (the number of chromosomes in the population). In essence, the size of the population depends greatly on the method used to create the initial population.

Moreover, as the size of the population increases, the complexity and memory usage increases, but on the other hand, the tendency to converge to a global optimum solution also increases.

Thus, it is required to have an initial population that has the necessary information and gene possibilities for the GA to converge, without the huge amount of chromosomes in a population.

The initial population is generated by creating an initial solution using one of the simple phase unwrapping algorithms such as ‘Quality guided phase unwrapping algorithm’. The initial solution is approximated using a ‘polynomial Surface-fitting weighted least-square multiple regression’ method. By multiple regression method, the initial coefficients of the polynomial will be generated. These coefficients are, then, inserted into the first chromosome of the genetic algorithm initial population.

This method for creating initial solution is quite powerful and gives the GA a good start to reach convergence. It is more intelligent and problem-specific than random initialization of polynomial coefficients. It also gives the GA the option of fewer chromosomes in a population and speeds up the convergence of the GA to an optimal solution.

### 3.3 Polynomial Surface-fitting Weighted Least-Square Multiple Regression

Surface-fitting using polynomials is a very well established subject used to best fit a polynomial to set of data. One way to surface fit a polynomial to a set of data is by weighted least-square multiple regression. This method of regression minimizes the sum of residuals (least square error or $L^2$-norm) controlled by a set of weights. It involves smoothing of the data or identifying an apparent trend in the data. The weights used define how good the data to be fitted is and how much they can contribute to the fitting of the polynomial to the data. The number of coefficients of the polynomial specifies the size of the matrices used to solve the least-square problem.

The best fitted polynomial surface has the minimum weighted least square error defined in Eq. (6):

$$S = \sum_{i=1}^{n} w_i [z_i - f(x_i, y_i)]^2$$

(6)

where $w_i$ is the weight at at pixel $i$. 
$z_i$ is the data pixel to be fitted, 
$f(x_i, y_j)$ is the pixel value evaluated using the polynomial in Eq. (5) at coordinate $x_i$ & $y_j$ in the x and y direction respectively.

The coefficients of the $n$-th order polynomial are the unknowns need to be evaluated to construct the surface. However, the known parameters are the data to be fitted $z_i$ and the two control points defined by the $x_i$ & $y_j$ coordinates of the polynomial surface in the x and y directions. To obtain the weighted least-square error, the unknown polynomial coefficients must yield zero first derivatives.

\[
\frac{dS}{da_0} = 2 \sum_{i=1}^{NM} w_i [z_i - f(x_i, y_j)] = 0
\]
\[
\frac{dS}{da_1} = 2 \sum_{i=1}^{NM} w_i [z_i - f(x_i, y_j)] = 0
\]
\[
\vdots
\]
\[
\frac{dS}{da_{(n+1)(n+2)/2}} = 2 \sum_{i=1}^{NM} w_i [z_i - f(x_i, y_j)] = 0
\]

By expanding Eq. (7) will result in the set of linear equations presented in Eq. (8):

\[
\begin{align*}
\sum_{i=1}^{NM} w^i + a_0 \sum_{i=1}^{NM} w^i x^i + a_1 \sum_{i=1}^{NM} w^i y^i &= \sum_{i=1}^{NM} w^i z^i \\
\sum_{i=1}^{NM} w^i x^i + a_0 \sum_{i=1}^{NM} w^i x^i y^i + a_1 \sum_{i=1}^{NM} w^i y^i &= \sum_{i=1}^{NM} w^i z^i \\
&\vdots \\
\sum_{i=1}^{NM} w^i y^i + a_0 \sum_{i=1}^{NM} w^i y^i y^i + a_1 \sum_{i=1}^{NM} w^i y^i &= \sum_{i=1}^{NM} w^i z^i \\
\end{align*}
\]

Eq. (8) can be simplified to the following matrix form:

\[
XA = Z
\]

where $A$ is the matrix representing the polynomial coefficients, 
is the matrix representing the left side of Eq. (8), 
$Z$ is the matrix representing the right side of Eq. (8).

By reordering the matrices, the coefficient matrix can be evaluated from Eq. (10):

\[
A = X^{-1}Z
\]

This can be solved by Gaussian Elimination to calculated the values of the coefficient of the best fit polynomial surface that can minimises the weighted least-square error.

### 3.4 Generating Initial Population Based on the Initial Solution

Once the initial solution coefficients are calculated, the coefficients of the initial solution are inserted in the first chromosome in the initial population. The rest of the population is generated using the following method: For every gene in the chromosome a random number is added to the value of the gene as in Eq. (11):

\[
a_i = a_i + \delta (\text{rand}-0.5)
\]

where $a_i$ is the coefficient parameter stored in gene ‘i’,
$\delta$ is the fraction number which limit the generated random number to be smaller than the value $a_i$,
Rand is a random number generated between the values [0, 1].

### 3.5 Fitness Evaluation

To find the global optimum solution to the parameter estimation $L^p$-norm phase unwrapping problem, the quality of the solution must be evaluated at every generation in order to inform the genetic algorithm of how good its current solution is at each stage. The evaluation will increase the knowledge of the GA of how good the quality level of the solution. This can be achieved by using a problem-specific fitness function specified in Eq. (1). The genes of a chosen chromosomes are substituted as coefficients in Eq. (2) to evaluate the approximated phase value at coordinate (i, j). The obtained phase is subtracted from the adjacent pixel approximated phase value to calculate the approximated unwrapped phase solution gradient. It is then subtracted from the gradient of the wrapped phase in the x and y direction. Then, the error is evaluated in the $L^p$-norm sense.

\[
\psi_{ij} = f(i, j) = a_0 + a_1 f + a_2 f + \ldots + a_{(n+1)(n+2)} f^n
\]

where $a_k$ are the parameter coefficient that will be estimated by the genetic algorithm to approximated the unwrapped phase that can achieve the minimum $L^p$-norm, i and j are indices of the pixel location in the image respectively.
3.6 Selection Operator

The selection operator is an important step in a genetic algorithm. This reproduction operator selects the fittest chromosomes for the current population and copies them to the new chromosomes in the next generation. It applies the natural concept of evolution, which states that: “the most fit individual survives to the next generation” [Willi-Hans et al., 2002]. Selected parent chromosomes must be suitable for crossover (mating) to generate new child chromosomes, i.e., new solutions that have a high tendency to be better (more fit) than their parent chromosomes. The selection operator is required to be intelligent and problem-specific in order to speed convergence and to avoid trapping the solution in local minima. It is required that the selection operator avoids causing a loss of population diversity and also avoids the ineffective execution of crossover operation [Willi-Hans et al., 2002].

3.7 Greedy 2-point Crossover

Crossover is a natural operator used to generate new chromosomes from original chromosomes. It is an important operator in genetic algorithm because it introduces diversity into chromosomes. It is capable of combining schemas (important genes) located in the original chromosomes. It avoids destroying the schemas in the original chromosomes, in essence, good schemas are conserved and propagated into the new chromosomes. Thus, it has more tendencies to create a better chromosome than weak one. A greedy method was also used in this crossover operator to ensure only best fit chromosomes are allowed to propagate into the new generation.

The greedy 2-point crossover is summarized as follows:

- Choose two chromosomes from the initial population for mating with a crossover probability; $P_x$,
- Choose randomly two crossover points,
- Swap the genes lying between these two points of one the chromosomes with the genes lying between the same points in the other chromosome; and vice versa,
- Evaluate the fitness of both new generated chromosomes,
- Compare the fitness of the new generated chromosomes with that of the original chromosomes,
- Choose the two best fit chromosomes from the original and the new chromosomes to be added into the new population.

Figure 4: The basic steps in the greedy 2-point crossover.

3.8 Mutation Operator

Mutation operator applies deliberate changes to a gene at random, to keep variation in genes and increasing the probability of not falling into a local minimum solution. It involves exploring the search space for new better solutions. This proposed operator uses a greedy technique which ensures only the best fit chromosome is allowed to propagate to the next generation.

This operator can be summarized as follows:

- Choose a chromosome from the current population at a probability; $P_m$,
- For every gene in the chromosome a random number is added to the value of the gene using Eq. (11).

3.9 Phase Matching

This method matches the phase of the wrapped phase with approximated unwrapped phase to establish the best representation of the unwrapped phase. The phase matching step extracts the small details embedded in the wrapped phase data which was lost in the global phase unwrapping. This step is performed using Eq. (13).

$$
\xi(p) = \Phi(p) + 2\pi \rho \left[ \frac{1}{2\pi} \left( \Phi(p) - \Psi(p) \right) \right]
$$

Where $\xi(p)$ is the phase matched unwrapped phase,
$\Phi(p)$ is the given wrapped phase,
$\Psi(p)$ is the approximated unwrapped phase,
$\rho(\cdot)$ is a rounding function defined by
$\rho[t] = \lfloor t + \frac{1}{2} \rfloor$ for $t \geq 0$ and
$\rho[t] = \lfloor t - \frac{1}{2} \rfloor$ for $t < 0$.
4 Results and Discussion

The proposed algorithm is tested on two kinds of wrapped phase maps; simulated and real phase maps, to verify the performance of the proposed algorithm. The results were also compared with a very well known global phase unwrapping algorithms developed by Ghiglia et al. called ‘L^p-norm two-dimensional phase unwrapping algorithm’. The results of all these stated algorithms were executed on a Pentium IV- 3.0 GHz computer. The order of the polynomial fitted was chosen experimentally depending on the object surface complexity.

4.1 Computer Simulated Results

The proposed algorithm was tested on computer simulated object with high noise and the result was compared with that of the L^p-norm algorithm and the weighted least squares algorithm. The wrapped phase map and the rewrapped result of all the algorithms are presented in Fig. 5. The proposed algorithm best matches the original wrapped phase with an advantage of smoothing the noise embedded in the wrapped phase map. The unwrapped surface is presented in Fig. 6.

![Figure 5](image1.png)

Figure 5: The simulated noisy object 128x128 (a) wrapped phase map, rewrapped phase map using (b) L^p-norm algorithm, (c) weighted least-square algorithm, (d) the proposed algorithm.

4.2 Experimental Results

The proposed algorithm was also implemented on a real wrapped phase map generated from Interferometric Synthetic Aperture Radar (IFSAR) data [Ghiglia et al., 1998]. The IFSAR wrapped phase map and the rewrapped result of the L^p-norm and the proposed algorithm are presented in Fig. 7.

![Figure 6](image2.png)

Figure 6: Simulated noisy object 128x128 (a) original 3d-surface, unwrapped phase map using (b) L^p-norm algorithm, (c) weighted least-square algorithm, (d) the proposed algorithm.

![Figure 7](image3.png)

Figure 7: (a) a 512x512 noisy IFSAR wrapped phase, rewrapped phase map using (b) L^p-norm algorithm, (c) weighted least-square algorithm, (d) the proposed algorithm.
Figure 8: 3D-surface of the unwrapped phase map for the noisy IFSAR wrapped phase map in Fig. 7(a) achieved using (a) Lp-norm min-grad, (b) weighted least-square algorithm and (c) the proposed algorithm.

The proposed algorithm proves to be very robust in unwrapping the wrapped phase map which when rewrapped achieves the best matching rewrapped phase map with that of the original except of the areas which was masked out by the quality map due to high noise.

Conclusion

A hybrid genetic algorithm using a parametric method to solve the two-dimensional phase unwrapping problem has been proposed. The proposed algorithm has been demonstrated to be very robust and computationally efficient. This algorithm uses a genetic algorithm to estimate the coefficient of a n-th order polynomial that best approximate the unwrapped phase map which minimizes the difference between the unwrapped phase gradient and the wrapped phase gradient. The genetic algorithm in this proposed method uses an initial solution to speed convergence.

The initial solution is achieved by unwrapping using a simple unwrapping algorithm and estimating the parameters of the polynomial using weighted least squares multiple regression. The parameters of the initial solution are then given to the genetic algorithm as an initial solution.

The algorithm was then tested on simulated and experimental data and it proved to be efficient and robust. The comparison of performance of this algorithm was made with powerful established phase unwrapping algorithms such as the Lp-norm. Based on the rewrapping of the solution, the proposed gave better result best matched the original wrapped phase map.

Reference


